

UNITED STATES PATENT APPLICATION

OF

NABIL N. SEDDIGH

BISWAJIT B. NANDY

DON W. BENNETT

YAJUN LIU

DABIN WANG

&

CARL F. CAO

FOR

**CONFIGURABLE RULE-ENGINE FOR LAYER-7
AND TRAFFIC CHARACTERISTIC-BASED CLASSIFICATION**

Prepared by:

**PILLSBURY MADISON & SUTRO LLP
INTELLECTUAL PROPERTY GROUP**

DATED: Nov. 24, 2000

CONFIGURABLE RULE-ENGINE FOR LAYER-7 AND TRAFFIC CHARACTERISTIC-BASED CLASSIFICATION

5

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention generally relates to the field of data communication systems. More particularly, the invention presents a configurable and extensible rule-engine capable of classifying and supporting packet traffic.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100

2. Description of Related Art and General Background

The unprecedented growth of the Internet has not only increased the amount of traffic that communication networks must support, it has also transformed the nature of network traffic. The Internet was once relegated to handling Internet Protocol (IP)-based transmissions in the form of Telnet, e-mail, and File Transfer Protocol (FTP) traffic originating from wired LAN/WAN networks. Since then, the Internet has evolved into a global information infrastructure capable of accommodating a wide variety of applications, such as World-Wide Web (WWW), Voice-over-IP (VoIP) and Audio/Video Playback, generated from a diverse set of media, including satellite, wireless, and optical platforms.

Presently, IP-based networks attempt to accommodate the traffic generated by such applications by providing a “best-effort” level of service. As such, all IP data packets must compete for available bandwidth, as well as processing capability and buffer space in the network routing devices. The use of IP-based applications with real-time/interactive requirements coupled with the relatively limited bandwidth capacity in access and wireless

networks precipitates the need to differentiate between the different traffic flows generated by these applications.

To this end, networks have incorporated classification mechanisms to differentiate among the various traffic flows flowing through the network. These mechanisms employ 5 packet classification schemes to help identify which data packets receive the necessary treatment to ensure a best-effort level of service.

A common approach implemented by these classification mechanisms is to classify the traffic into a static set of coarse traffic classes based on certain application requirements. Based on these coarse traffic classes, network routing devices provide 10 differentiated treatment.

Currently, some classification mechanisms perform packet classification based on Layer-4 (Transfer Control Protocol (TCP)/User Datagram Protocol (UDP)) Port Numbers. Although relatively simple to implement, such classification may be easily deceived by users manipulating port numbers to achieve higher levels of priority for applications.

15 Moreover, the use of port numbers in applications, although well-known, are not mandatory, thereby compromising the efficacy of the Layer-4 classification schemes. In addition, many networks employ Internet Protocol Security (IPSec) techniques, which provide for the secure exchange of packets, but do so at the expense of encrypting information above Layer-3 (Network Layer), thus rendering Layer-4 classification futile.

20 Other classification mechanisms support packet classification based on Layer-7 (Application Layer) content. Such classification schemes exploit the payload information resident in the data packet to better identify the type of application associated with the traffic and overcomes the limitations of Layer-4 classification schemes noted above. However, Layer-7 classification schemes require a larger, more robust set of rules to

operate effectively and is still subject to the classification barriers imposed by IPSec techniques.

Recent efforts, as described in Chapman *et al.*, *Automatic Quality of Service in IP Networks*, PROC. CANADIAN CONF. ON BROADBAND RESEARCH, Ottawa, Canada (April 5 1997, pp. 184-189), have investigated the use of flow classification schemes, which classifies flows based on traffic characteristics. Dynamic flow classification schemes examine certain flow qualities, such as, for example, transmitted packet counts and inter-arrival times, to determine the class associated with the traffic flow. The set of rules associated with dynamic flow classification schemes are, therefore, proportional to the number of classes in the classification scheme. As such, the rule set maintained by these schemes are smaller than the Layer-7 classification schemes. Moreover, because, dynamic flow classification schemes examine traffic flow behavior, such schemes may overcome the classification barriers imposed by IPSec techniques.

As noted above, conventional Layer-7 classification mechanisms employ a static set of rules, which differentiate network traffic into coarse traffic classes. Typically, these rules are hard-coded into the classification mechanisms, thus precluding network administrators from readily extending, configuring, or modifying the rules. Such rules limit the recognition and classification of traffic generated by the wide variety of applications currently supported by networking devices.

20 Therefore, what is needed is a system and method that provides a greater flexibility in classifying network traffic.

SUMMARY OF INVENTION

Systems and methods consistent with the principles of the present invention, as embodied and broadly described herein, include a data flow managing mechanism configured to identify, track, and manage a data flow and a rule set, which includes a plurality of rules for comparing information contained within the data flow to pre-specified values. The system further includes a configurable classification rule engine for classifying the data flow into one of a plurality of traffic classes based on results of the comparisons between the rules and the pre-specified values. The configurable classification rule engine is configured via a configuration file that specifies and allows for the modification and reconfiguration of the pre-specified values and information regarding the data flow, the rule set, and the traffic classes.

BRIEF DESCRIPTION OF THE DRAWINGS

15 FIG. 1A depicts a functional block diagram of a classification rule engine system, constructed and operative in accordance with an embodiment of the present invention.

FIG. 1B depicts a data structure for a traffic monitoring mechanism, in accordance with an embodiment of the present invention.

FIG. 1C illustrates a functional state diagram for TCP-based traffic.

20 FIG. 1D illustrates a function state diagram for UDP-based traffic.

FIG. 1E depicts a data structure for a packet classification rule engine, constructed and operative in accordance with an embodiment of the present invention.

FIG. 2A depicts a functional block diagram of a classification rule engine system, constructed and operative in accordance with another embodiment of the present invention.

FIG. 2B depicts a data structure for a packet classification rule engine, constructed 5 and operative in accordance with another embodiment of the present invention.

FIG. 3 illustrates a configuration file for a packet classification rule engine in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The following detailed description refers to the accompanying drawings that illustrate embodiments of the present invention. Other embodiments are possible and modifications may be made to the embodiments without departing from the spirit and scope of the invention. Therefore, the following detailed description is not meant to limit the invention.

15 Rather the scope of the invention is defined by the appended claims.

The classification rule system, as described herein, is a configurable rule-based system capable of classifying network traffic based on application requirements. The system employs classification rules that are abstracted from the heuristics and are not hard-coded. In one embodiment, the classification system operates under a dynamic 20 classification scheme, employing a traffic monitoring mechanism to monitor and measure certain traffic characteristics from a received data flow and compare those values to the rules invoked by the classification rule engine to designate the various classes. In another embodiment, the system operates under a Layer-7 classification scheme and examines predetermined packets containing certain application information to determine whether

certain patterns, indicative of different classes, match the content of the predetermined packets.

FIG. 1A is a functional block diagram depicting dynamic classification rule engine system 100, constructed and operative in accordance with an embodiment of the present invention. System 100 classifies network traffic based on dynamic classification schemes. As indicated in FIG. 1A, system 100 comprises flow manager mechanism 102, a traffic monitoring mechanism 106, a packet classification rule engine 108, configuration file 300, and rule set 110. As will be described below, traffic monitoring mechanism 106 and rule set 110, as well as the actions taken by packet classification rule engine 108, may be defined, initialized, and reconfigured with respect to particular traffic flows in configuration file 300.

Flow manager mechanism 102 is configured to receive the incoming data packets, identify and track the traffic flows associated with the received packets, register active data flows, and delete inactive data flows. For dynamic classification schemes, flow manager mechanism 102 may identify the traffic flows based on Source IP Address (SIP), Destination IP Address (DIP), Source TCP/UDP Port Number (SP), Destination UDP/TCP Port Number (DP) and Protocol Id (PID). Flow manager mechanism 102 may include a flow table mechanism 104, which captures the relevant packet information, provides a look-up mechanism to identify which traffic flow the receive packet belongs to, and updates the captured information accordingly. Such updates may include introducing new traffic flows or deleting stale traffic flows.

Traffic monitoring mechanism 106 is configured to measure various characteristics and parameters of each traffic flow and provides updates to the flow information resident in flow table mechanism 104, whenever a received packet is associated with a particular

flow. Traffic monitoring mechanism 106 may comprise a plurality of individual monitors dedicated to measuring the various traffic flow characteristics and parameters. An exemplary sample of individual monitors are summarized in TABLE I.

5

TABLE I

TMName	TMTType	TMDescription
CPLC	0	number of consecutive packets since last state change
CLP	1	number of consecutive long packets having a size \geq LONG
CSP	2	number of consecutive short packets having a size \leq SHORT
AVG_BW	3	average bandwidth measured over time interval T
PPS	4	average packets-per-second measured over time interval T
I_AVG	5	average packet inter-arrival time I_{avg}
PIGA	6	number of packets with inter-arrival time greater than A = $(I_{avg} + Y*(I_{avg}))$
PILB	7	number of packets with inter-arrival time less than B = $(I_{avg} - Y*(I_{avg}))$
PS	8	number of packets since start of the flow
DIFI	9	inter-arrival time between current packet and previous packet in flow
TLP	10	time of last packet in flow
PS_AVG	11	average packet size measured over time interval T
SCP	12	size of current packet (includes entire packet)
SCPL	13	size of IP payload for current packet

In one implementation, the values for indicating the size of a long and short packet (LONG, SHORT, respectively), as used in monitors CLP and CSP noted above, as well as 10 time interval T are user-definable.

FIG. 1B depicts the data structure for maintaining an easily extensible set of monitors in traffic monitoring mechanism 106. The traffic monitor data structure is chained off flow table 104. Although in the illustrated embodiment, flow table 104 is depicted as a linked list, flow table 104 may utilize any data structure that maintains the 15 traffic monitor data structure. As indicated in FIG. 1B, each flow entry 104A-J of flow table 104 points to a traffic monitor data structure 106A-J having identifiers "TRAFFICMONITORIndex" to identify the traffic monitors. In turn, traffic monitor data structure 106A-J contains at least one entry for every type of traffic monitor supported.

Although each monitor type has its own corresponding data structure **106A1-J1**, all monitor types share a common set of fields. This common set of fields may be configured to include: "TMName," which identifies the monitor via an identifier string (see, TABLE I); "TMType," which identifies the monitor type (see, TABLE I); "TMValue," which provides the value of an observed parameter, as detected by corresponding monitor; and "SCRAPARRAY(•)," which provides a scrap area, the usage of which will vary depending on the requirements of the particular monitor.

Rule set **110** includes a plurality of rules designed to perform comparisons between traffic monitoring mechanism **106** and a configured or pre-specified value. As such, rule set **110** may include event indicia, indicating the triggering of a rule, condition indicia, representing a conditional or comparison statement, and action indicia, indicating the execution of an action based on the results of the comparison. Rule set **110** may employ comparison operators, such as, GREATER THAN, GREATER THAN OR EQUAL TO, EQUAL TO, LESS THAN OR EQUAL TO, and LESS THAN, for example, to effect the comparisons. Rule set **110** may also employ logical operators, such as, AND, OR, NAND, NOR, etc. to chain various rules together. As will be discussed below, rule set **110** may be defined in accordance with rule data structures **158A**, **158B**, **168A**, **168B**. Based on the results rendered by the application of the rule set **110** to the received traffic flow, packet classification rule engine **108** reacts to ensure that the received traffic is associated with the appropriate class.

Before describing packet classification rule engine **108**, it may be instructive to generally describe an example of a dynamic classification scheme for two types of traffic commonly found in IP-based networks, TCP and UDP traffic. As is well known, TCP is a connection-oriented protocol, which guarantees the delivery and sequential order of the

transmitted data. UDP is a connection-less protocol, offering little in the way of delivery guarantees, but provides a more direct means of sending and receiving data. Further, it is to be noted that the depicted dynamic classification schemes, and the rule sets used in effecting transitions between classes, are for illustrative purposes and other classification 5 schemes and rule sets may be implemented without departing from the thrust of the present invention.

FIG. 1C depicts a classification state diagram 125 as well as the TCP rule set for the dynamic classification scheme applied to TCP traffic. Interactive Query Response (IQR) class 126 represents traffic comprising a small number packets that need to be transmitted with high priority and little delay due to the interactive nature of the application (e.g., voice over IP, WWW, e-commerce applications, etc.). Burst class 128 represents traffic comprising a medium number of packets that are transmitted in an infrequent and bursty manner. Bulk class 130 represents traffic comprising a large number packets that are capable of tolerating delays due to large volume transfers (e.g., 10 Bulk class 130 represents traffic comprising a large number packets that are capable of tolerating delays due to large volume transfers (e.g., 15 File Transfer Protocol (FTP) traffic).

As indicated in FIG. 1C, the dynamic classification scheme defaults all new TCP flows to IQR class 126. In accordance with rule **R**₀, if no new TCP packet is detected by traffic monitoring mechanism 106 within one second, IQR class 126 transitions to itself. In accordance with rule **R**₁, if traffic monitoring mechanism 106 detects at least two 20 consecutive TCP packets having a size greater than or equal to LONG (e.g., 512 bytes), IQR class 126 transitions to Burst class 128. Such a rule may be implemented in order to adequately track and classify consecutive long packets having an occasional short packet for separation. In accordance with rule **R**₂, upon traffic monitoring mechanism 106 detecting ten consecutive TCP packets having a size greater than or equal to LONG, Burst

class **128** transitions to Bulk class **130**. Finally, In accordance with rule **R₃**, if traffic monitoring mechanism **106** does not detect any TCP packets within one second, Bulk class **130** transitions back to IQR class **126**.

FIG. 1D depicts a classification state diagram **125** as well as the UDP rule set for the exemplary dynamic classification scheme applied to UDP traffic. Low Rate Real Time (LRT) class **126** represents traffic comprising a small number packets that need to be transmitted in real time. High Rate Real Time (HRT) class **128** represents traffic comprising a large number of packets that need to transmitted in real time. Non-Real Time (NRT) class **130** represents traffic that does not require real time transfers.

As indicated in FIG. 1D, the dynamic classification scheme defaults all new UDP flows to LRT class **136**. In accordance with rule **R₀**, if traffic monitoring mechanism **106** does not detect any packets for a certain interval of time (e.g., 1 sec.), LRT class **136** transitions to HRT class **140**. In accordance with rule **R₁**, upon traffic monitoring mechanism **106** detecting a uni-modal inter-arrival time distribution (e.g., flowing video packets), HRT class **138** transitions to NRT class **140**. Finally, in accordance with rule **R₂**, if traffic monitoring mechanism **106** detects transmissions of at least 25 UDP packets per second and available bandwidth of at least 50 Kbps, NRT class **140** transitions back to LRT class **136**.

FIG. 1E depicts a data structure **150** for packet classification rule engine **108**. Packet classification rule engine **108** performs actions to classify the received packets into their proper class in response to the information provided by flow manager mechanism **102** and traffic monitoring mechanism **106**. To this end, classification rule engine data structure **150** enables a user to specify a configurable number of classes, as well as transitions between those classes, based on a configurable set of rules and actions. For

purposes of illustration, data structure 150 depicted in FIG. 1E is based on the dynamic classification schemes 125, 135 developed above for TCP and UDP traffic, respectively. Classification rule engine data structure 150 may be configured to accommodate any type of IP-based traffic. In the illustrated embodiment, rule engine data structure 150 5 contemplates the dynamic classification of TCP and UDP traffic by employing classification machines that relate to the two traffic types: a TCP classification machine data structure 152 and a UDP classification machine data structure 162. Each classification machine data structure 152, 162 may correspond to three or more classes. For example, TCP classification machine data structure 152 may correspond to IQR class 10 126, Burst class 128, and Bulk class 130. Similarly, UDP classification machine data structure 162 may contain LRT class 136, HRT class 138, and NRT class 140. The invention is not limited in this respect, however, as more or less classes may be implemented.

As indicated in FIG. 1E, classification machine data structures 152, 162 may share 15 a common set of fields. This common set of fields may be configured to include: "ClassName," which identifies the type of traffic accommodated by the data structure and "ClassIndex," which identifies the index number associated with the data structure. Upon determining which classification machine 152, 162 the packets belong to, classification rule engine data structure 150 may employ a pointing mechanism, such as, 20 "CLASSPointer" to point to traffic class data structures 154, 164 corresponding to the different classes of traffic.

Traffic class data structures 154, 164 may share a common set of fields. This common set of fields may be configured to include: "ClassIndex," which identifies the number assigned to the class; "ClassName," which identifies the name of the class; and

“Num_Transitions,” which identifies the number of transitions each class is capable of executing. As indicated in FIG. 1E, traffic class data structure **154** for TCP classification machine **152** includes ClassIndex = “0,” “1,” and “2” and ClassName = “IQR,” “Burst,” and “Bulk” corresponding to the respective three classes of TCP traffic noted above and

5 Num_Transitions = “2,” “1,” “1” corresponding to the different transitions possible in each of the respective classes (see also, FIG. 1D).

Similarly, traffic class data structure **164** for UDP classification machine **162** includes ClassIndex = “0,” “1,” and “2” and ClassName = “LRT,” “HRT,” and “NRT” corresponding to the respective three classes of UDP traffic noted above and

10 Num_Transitions = “1,” “1,” “1” corresponding to the different transitions possible in each of the respective classes (see also, FIG. 1D). In addition, traffic class data structures **154**, **164** may also include a pointing mechanism, such as, “TransitionPtr” to point to transition data structures **156A**, **156B**, **166**, corresponding to the different transitions possible for each class of traffic.

15 Transition data structures **156A**, **156B**, **166**, may share a common set of fields. This common set of fields may be configured to include: “TransitionIndex,” which identifies the transition executed by a particular class and “NextRulePtr,” which identifies the rule associated with the transition executed by the class. If a particular class is capable of executing two transitions (e.g., Num_Transitions = 2), transition data structures **156A**,

20 **156B**, **166** may implement a pointing mechanism, such as, “NextTransitionPtr,” to point to and facilitate the second transition if the conditions for the first transition were not met (i.e., no rule match). If there is no subsequent transitions, NextTransitionPtr points to a NULL state. For example, as indicated in FIG. 1E, the first of two transition possibilities of IQR class **126** (i.e., ClassIndex = 0 and ClassName = “IQR”), is denoted as

TransitionIndex = 0. As illustrated in FIG. 1C, this transition occurs if no TCP packets are received for 1 sec. (see, Rule **R₁**). If the conditions of the rule associated with NextRulePtr of TransitionIndex = 0 are not satisfied (i.e., Rule **R₁** is not matched), NextTransitionPtr points to the next transition, TransitionIndex = 1, which occurs if at 5 least 2 consecutive long packets are received (see, Rule **R₄**). If Rule **R₄** is not matched, NextTransitionPtr points to the NULL state because no other transitions are supported.

As noted above, "NextRulePtr" identifies the rule precipitating the transition executed by the class. As indicated in FIG. 1E, the rules are defined in rule data structures **158A**, **158B**, **168A**, **168B**. Rule data structures **158A**, **158B**, **168A**, **168B** may share a common set of fields. This common set of fields may be configured to include: "MonitorTypeIndex," which identifies the type of traffic monitor used by the rule; "RuleType," which identifies the comparison operators used by the rule; "RuleValue," which represents the value the rule compares to the values observed by the traffic monitor; "YesMatchClassIndex," which identifies the class to be designated based on a rule match; "NextRuleConnect," which indicates the logical operator used to chain to another rule before transitioning classes; and "NextRulePtr," which identifies the next rule to be examined before transitioning classes.

By way of example, consider transition data structures **156A**, **156B**. As indicated in FIG. 1E, transition data structure **156A** stems from IQR class **126** (i.e., ClassIndex = 0), 20 which is capable of two transitions (i.e., Num_Transitions = 2), due to compliance with Rules **R₀** or **R₁** (see, FIG. 1C). Because the rules (and transitions) are executed sequentially, transition data structure **156A** first points to corresponding rule data structure **158A**, which represents Rule **R₀**. Rule data structure **158A** provides that the rule employs a DIFI monitor (i.e., MonitorTypeIndex = 9), which observes the inter-arrival time

between a current packet and a previous packet. Rule data structure **158A** further provides that the rule employs the comparison operator (i.e., RuleType) GREATER THAN, and the comparison value (i.e., RuleValue) 0, thereby indicating that the inter-arrival time observed by the DIFI monitor has to be greater than 1 predetermined time interval (e.g., 5 μ sec., msec., sec., etc.), in order for IQR class **126** to transition back to itself (i.e., YesMatchClassIndex = 0, see also, FIG. 1C).

If the rule manifested by rule data structure **158A** is not matched by the observed traffic, transition data structure **156A** then points, by virtue of NextTransitionPtr, to transition data structure **156B**, which points to rule data structure **158B** representing Rule **R₁**. Rule data structure **158B** provides that the rule employs a CLP monitor (i.e., MonitorTypeIndex = 0), which observes the number of consecutive LONG packets. Rule data structure **158A** further provides that the rule employs the comparison operator (i.e., RuleType) GREATER THAN or EQUAL TO, and the comparison value (i.e., RuleValue) 2, thereby indicating that the number observed by the CLP monitor has to be greater than or equal 2 packets in order to effect a transition to Burst class **128** (i.e., YesMatchClassIndex = 1, see also, FIG. 1C).

With respect to rule data structures **168A**, **168B**, FIG. 1E also demonstrates the capability of chaining of rules noted above. Transition data structure **166** stems from NRT class **140** (i.e., ClassIndex = 2), which is capable of a single transition (i.e., 20 Num_Transitions = 1), due to compliance with Rule **R₂** (see, FIG. 1D). Transition data structure **156A** points to corresponding rule data structure **168A**, which represents Rule **R₂**. Rule data structure **168A** provides that the rule employs a PPS monitor (i.e., MonitorTypeIndex = 5), which calculates the average packets-per-second. Rule data structure **168A** further provides that the rule employs the comparison operator (i.e.,

RuleType) GREATER THAN, and the comparison value (i.e., RuleValue) 25, thereby indicating that the number observed by the PPS monitor must be greater than 25 packets in order to effect a transition to LRT class 136 (i.e., YesMatchClassIndex = 0, see also, FIG. 1D).

5 However, rule data structure 168A also provides that there is an additional condition (or rule) that must be complied with before completely transitioning into LRT class 136. Specifically, rule data structure 168A includes the string, NextRuleConnect = AND, indicating that that rules or conditions manifested by rule data structure 168A are to be chained to a subsequent rule or condition indicated by NextRulePtr. NextRulePtr points to rule data structure 168B, which provides that the subsequent rule or condition employs a AVG_BW monitor (i.e., MonitorTypeIndex = 4), which determines the average bandwidth. Rule data structure 168B also provides that the rule employs the comparison operator (i.e., RuleType) GREATER THAN, and the comparison value (i.e., RuleValue) 50, thereby indicating that the bandwidth observed by the AVG_BW monitor must be greater than 50 Kbps in order to completely match the transition rule and effect a transition to LRT class 136 (i.e., YesMatchClassIndex = 0, see also, FIG. 1D).

FIG. 2A is a functional block diagram depicting Layer-7 classification system 200, constructed and operative in accordance with another embodiment of the present invention. System 200 classifies network traffic based on Layer-7 classification schemes. 20 As indicated in FIG. 2A, system 200 comprises flow manager mechanism 202, a packet classification rule engine 208, a configuration file 300, and rule set 210. As will be described below, rule set 210, as well as the actions taken by packet classification rule engine 208, may be defined, initialized, and reconfigured with respect to particular traffic flows in configuration file 300.

Much like the flow manager mechanism of system 100, flow manager mechanism 202 is configured to receive the incoming data packets, identify and track the traffic flows associated with the received packets, as well as monitor the traffic flow statistics. Because Layer-7 classification schemes operate on packet payload information, only a few packets 5 within a traffic flow need to be examined to correctly identify the class of traffic. As such, flow manager mechanism 202 may identify the traffic flows by examining a predetermined packet or packets per flow. Flow manager mechanism 202 may include a flow table mechanism 204, which captures the relevant packet information, provides a look-up mechanism to identify which traffic flow the receive packet belongs to, and updates the captured information accordingly. Such updates may include introducing new traffic flows, deleting stale traffic flows, or including pointers to different flows.

INSAU 15
10 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 9999 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10099 10100 10101 10102 10103 10104 10105 10106 10107 10108 10109 10110 10111 10112 10113 10114 10115 10116 10117 10118 10119 10120 10121 10122 10123 10124 10125 10126 10127 10128 10129 10130 10131 10132 10133 10134 10135 10136 10137 10138 10139 10140 10141 10142 10143 10144 10145 10146 10147 10148 10149 10150 10151 10152 10153 10154 10155 10156 10157 10158 10159 10160 10161 10162 10163 10164 10165 10166 10167 10168 10169 10170 10171 10172 10173 10174 10175 10176 10177 10178 10179 10180 10181 10182 10183 10184 10185 10186 10187 10188 10189 10190 10191 10192 10193 10194 10195 10196 10197 10198 10199 10200 10201 10202 10203 10204 10205 10206 10207 10208 10209 10210 10211 10212 10213 10214 10215 10216 10217 10218 10219 10220 10221 10222 10223 10224 10225 10226 10227 10228 10229 10230 10231 10232 10233 10234 10235 10236 10237 10238 10239 10240 10241 10242 10243 10244 10245 10246 10247 10248 10249 10250 10251 10252 10253 10254 10255 10256 10257 10258 10259 10260 10261 10262 10263 10264 10265 10266 10267 10268 10269 10270 10271 10272 10273 10274 10275 10276 10277 10278 10279 10280 10281 10282 10283 10284 10285 10286 10287 10288 10289 10290 10291 102

using a packet number as an index into the data structure. As such, the rules corresponding to a particular packet number are searched to determine a match.

In the illustrated embodiment, packet numbers (e.g., #i, #j) within flow A are used as an index. Designating packet numbers #i, #j as an index may be based on the 5 knowledge that, under Layer-7 schemes, packet numbers #i, #j provide an indication as to the type of traffic in flow A. Such knowledge may be garnered from empirical data, heuristics, vendor specifications, or industry knowledge. Classification rule engine 208 then searches the set of rules 210A-D corresponding to packet numbers #i, #j to determine if there is a match between the information contained in the pointed-to packets and the rule 10 values.

As noted above, Layer-7 classification rule engine 208 is constructed as linked list of pattern match rules 210A-D. Rules 210A-D are designed to perform comparisons between the flows configured or pre-specified value. Rules 210A-D may include event 15 indicia, indicating the triggering of a rule, condition indicia, representing a conditional or comparison string, and action indicia, indicating the execution of an action based on the results of the comparison. Rules 210A-D may also employ comparison operators, such as, GREATER THAN, GREATER THAN OR EQUAL TO, EQUAL TO, LESS THAN OR 20 EQUAL TO, and LESS THAN, for example, to effect the comparisons. Rules 210A-D may also employ logical operators, such as, AND, OR, NAND, NOR, etc. to chain various rules together.

As such, classification rule engine data structure 250 may employ pattern match rules 210A-D having a common set of fields. This common set of fields may be configured to include: "Offset," which specifies the number of bytes from the beginning of the pointed-to packets where traffic type information is located; "NumCharacters," which

indicates the number of characters, beginning at the specified Offset, that are to be examined and compared for pattern matching; "CLASSPattern," which identifies the pattern used to match the information contained in the pointed-to packet with a particular class; and "AppType," which identifies the particular class associated with the pointed-to
5 packet.

Returning to FIG. 2B, with respect to packet #i, classification rule engine data structure 250 points to packet #i of flow A to seek traffic type information. Upon accessing packet #i, classification rule engine data structure 250 searches out all the relevant pattern matching rules for packet #i, which for this case are rules 210A-C.

Classification rule engine data structure 250 then sequentially applies the relevant pattern matching rules to determine a match. For example, classification rule engine data structure 250 applies rule 210A, which compares the x number of characters beginning at Offset a of packet #i with the pattern A. If the comparison with rule 210A does not consummate in a match, classification rule engine data structure 250 skips to rule 210C for subsequent comparisons. If any of the comparisons result in a match, classification rule engine data structure 250 designates the class of the traffic flow (e.g., AppType) based on the matched pattern (e.g., CLASSPattern). A flag (not shown) is then set in the Flow Table 204, indicating that the entire traffic flow (e.g., traffic flow A) is to be designated as traffic class AppType and future packets belonging to the traffic flow will not be examined. If no comparisons result in a match, classification rule engine data structure 250 maintains the traffic class the same and informs Flow Table 204 to continue examining the data structure in accordance with the next packet number that may have an indication as to the type of traffic in the flow.

Classification rule engine data structure 250 also provides for the chaining of rules to subsequent rules. The chaining of rules is contemplated because certain application identification may be based on heuristics, which may require pattern matching in two or more successive packets. When all the rules are traversed, there is only the pointed-to 5 packet in hand. Thus, if a successful match causes the transition to a chained rule for a subsequent packet, classification rule engine data structure 250 stores a pointing mechanism to the chained rule in Flow Table 204. All subsequent packets for that particular flow will no longer index off the packet-number. Instead, when the packet with the number corresponding to the transitioned chain rule arrives, classification rule engine 10 data structure 250 will compare the information contained in the packet to determine a match. If a match exists, classification rule engine data structure 250 will either proceed to the next rule (if it exists) or an application type is determined. If the match is not successful, application type and class designation remain unchanged.

For example, in the case where there exists a match in rule 210A, as indicated in 15 the illustrated embodiment (i.e., the x number of characters beginning at Offset a of packet #i contain pattern A), classification rule engine data structure 250 proceeds to rule 210B to determine if there is a subsequent match (i.e., do the y number of characters beginning at Offset b of packet #j contain pattern B). If there is a subsequent match, classification rule engine data structure 250 designates the traffic class as AppType = 1; otherwise, 20 classification rule engine data structure 250 designates the traffic class as the default class.

In this embodiment, all traffic flows are assumed to be assigned a default classification when they are initially received. This default application is based on the Layer-4 port numbers. As such, even if the Layer-7 Classification does not yield any

conclusive results, Layer-7 packet classification rule engine **208** falls back on the results of the Layer-4 port numbers to classify the traffic flows.

FIG. 3 depicts a packet classification rule engine configuration file **300**, constructed and operative in accordance with aspects of an embodiment of the present invention. Configuration file **300**, provides an easily modifiable interface to the classification rule engine, which enables a user to develop a classification algorithm, translate that algorithm into the set of rules supported by the rule-engine and enter the rules in the form of a simple grammar into a file. The classification rule-engine on power-up will execute the classification algorithms. A benefit of such a configuration file and classification rule-engine is that modifying and upgrading classification algorithms is a matter of simply making changes to a text file rather than providing hardware or software upgrades.

As indicated in FIG. 3, configuration file **300** provides an example grammar through which the classification rule engine may be configured as part of a bandwidth management system. In illustrated embodiment, configuration file **300** specifies the network device physical ports that are enabled and the bandwidth associated with each port. There is also a policy setting that allows the user to select the classification result from either a static layer-7 classifier or a dynamic classifier. Both types of classifiers run in parallel and their results are combined to obtain a final class.

The user is also allowed to specify the number of distinct classes and their names. In addition the scheduler type is configurable along with the scheduler parameters such as queue weights etc. For the Layer-7 classifier, the classification engine uses a rule to identify an application and then maps that application to a particular class. Finally, the grammar contains rules for transitioning between classes for the dynamic classifier.

It will be apparent to one of ordinary skill in the art that the embodiments as described below may be implemented in many different embodiments of software, firmware, and hardware in the entities illustrated in the figures. The actual software code or specialized control hardware used to implement the present invention is not limiting of 5 the present invention. Thus, the operation and behavior of the embodiments will be described without specific reference to the actual software code or specialized hardware components. The absence of such specific references is feasible because it is clearly understood that artisans of ordinary skill would be able to design software and control hardware to implement the embodiments of the present invention based on the description 10 herein.

Moreover, the processes associated with the presented embodiments may be stored in any storage device, such as, for example, non-volatile memory, an optical disk, magnetic tape, or magnetic disk. Furthermore, the processes may be programmed when the system is manufactured or via a computer-readable medium at a later date. Such a 15 medium may include any of the forms listed above with respect to storage devices and may further include, for example, a carrier wave modulated, or otherwise manipulated, to convey instructions that can be read, demodulated/decoded and executed by the system.

The foregoing description of the preferred embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to 20 these embodiments are possible, and the generic principles presented herein may be applied to other embodiments as well. For example, the invention may be implemented in part or in whole as a hard-wired circuit, as a circuit configuration fabricated into an application-specific integrated circuit, or as a firmware program loaded into non-volatile storage or a software program loaded from or into a data storage medium as machine-

readable code, such code being instructions executable by an array of logic elements such as a microprocessor or other digital signal processing unit.

As such, the present invention is not intended to be limited to the embodiments shown above but rather is to be accorded the widest scope consistent with the principles 5 and novel features disclosed in any fashion herein.

027015-142600